

Amendments to the Claims

1 Claim 1 (previously presented): A computer program product for sending Transmission Control
2 Protocol (TCP) messages through HyperText Transfer Protocol (HTTP) systems, the computer
3 program product embodied on one or more computer-readable media and comprising:

4 computer-readable program code means for establishing a send channel from a first
5 component on a client side of a network, through one or more HTTP-based systems, to a second
6 component on a remote side of the network;

7 computer-readable program code means for establishing a receive channel from the first
8 component, through the one or more HTTP-based systems, to the second component, wherein the
9 receive channel is distinct from the send channel;

10 computer-readable program code means for establishing a first TCP connection from a
11 client on the client side to the first component;

12 computer-readable program code means for establishing a second TCP connection from
13 the second component to a target server on the remote side;

14 computer-readable program code means for transmitting client-initiated TCP requests
15 from the client to the target server by packaging the client-initiated TCP requests into HTTP
16 messages which are transmitted on the send channel; and

17 computer-readable program code means for transmitting server-initiated TCP requests
18 from the target server to the client by packaging the server-initiated TCP requests into HTTP
19 messages which are transmitted on the receive channel.

1 Claim 2 (previously presented): The computer program product according to Claim 1, wherein

2 the computer-readable program code means for transmitting client-initiated TCP requests further
3 comprises:

4 computer-readable program code means for receiving a client-initiated TCP request from
5 the client at the first component on the first TCP connection;

6 computer-readable program code means for packaging the received client-initiated TCP
7 request in an HTTP POST request message;

8 computer-readable program code means for sending the HTTP POST request message to
9 the second component on the send channel;

10 computer-readable program code means for receiving the sent HTTP POST request
11 message at the second component;

12 computer-readable program code means for extracting the client-initiated TCP request
13 from the received HTTP POST request message; and

14 computer-readable program code means for forwarding the extracted client-initiated TCP
15 request to the target server on the second TCP connection.

1 Claim 3 (previously presented): The computer program product according to Claim 2, wherein
2 the computer-readable program code means for transmitting client-initiated TCP requests further
3 comprises computer-readable program code means for acknowledging the HTTP POST request
4 by sending an HTTP POST response from the second component to the first component on the
5 send channel.

1 Claim 4 (original): The computer program product according to Claim 3, wherein the computer-

2 readable program code means for establishing the send channel operates in response to the
3 computer-readable program code means for receiving the client-initiated TCP request, and
4 wherein the computer-readable program code means for transmitting client-initiated TCP
5 requests further comprises:

6 computer-readable program code means for receiving the HTTP POST response at the
7 first component; and

8 computer-readable program code means for closing the send channel, responsive to
9 operation of the computer-readable program code means for receiving the HTTP POST response.

1 Claim 5 (previously presented): The computer program product according to Claim 1, wherein
2 the computer-readable program code means for transmitting server-initiated TCP requests further
3 comprises:

4 computer-readable program code means for sending an HTTP GET request message from
5 the first component to the second component on the receive channel;

6 computer-readable program code means for receiving the sent HTTP GET request
7 message at the second component;

8 computer-readable program code means for receiving a server-initiated TCP request from
9 the target server at the second component on the second TCP connection;

10 computer-readable program code means for packaging the received server-initiated TCP
11 request in an HTTP GET response message which acknowledges the received HTTP GET
12 request message;

13 computer-readable program code means for sending the HTTP GET response message

14 from the second component to the first component on the receive channel;
15 computer-readable program code means for receiving the sent HTTP GET response
16 message at the first component;
17 computer-readable program code means for extracting the server-initiated TCP request
18 from the received HTTP GET response message; and
19 computer-readable program code means for forwarding the extracted server-initiated TCP
20 request to the client on the first TCP connection.

1 Claim 6 (original): The computer program product according to Claim 5, wherein the computer-
2 readable program code means for transmitting server-initiated TCP requests further comprises:
3 computer-readable program code means for performing a read operation on the second
4 TCP connection, responsive to operation of the computer-readable program code means for
5 receiving the sent HTTP GET request message and prior to operation of the computer-readable
6 program code means for receiving the server-initiated TCP request; and
7 computer-readable program code means for using the received server-initiated TCP
8 request as a result of the read operation, thereby triggering operation of the computer-readable
9 program code means for packaging the received server-initiated TCP request in the HTTP GET
10 response message.

1 Claim 7 (original): The computer program product according to Claim 5, wherein the computer-
2 readable program code means for transmitting server-initiated TCP requests further comprises
3 computer-readable program code means for preparing to receive another server-initiated TCP

4 request by triggering operation of the computer-readable program code means for sending the
5 HTTP GET request message from the first component to the second component, responsive to
6 operation of the computer-readable program code means for receiving the sent HTTP GET
7 response message at the first component.

1 Claim 8 (original): The computer program product according to Claim 2, wherein a Multi-
2 Purpose Internet Mail Extensions (MIME) type of the HTTP POST request message is set to
3 "binary/tcp".

1 Claim 9 (original): The computer program product according to Claim 5, wherein a Multi-
2 Purpose Internet Mail Extensions (MIME) type of the HTTP GET request message is set to
3 "binary/tcp".

1 Claim 10 (previously presented): A system for sending Transmission Control Protocol (TCP)
2 messages through HyperText Transfer Protocol (HTTP) systems, comprising:
3 means for establishing a send channel from a first component on a client side of a
4 network, through one or more HTTP-based systems, to a second component on a remote side of
5 the network;
6 means for establishing a receive channel from the first component, through the one or
7 more HTTP-based systems, to the second component, wherein the receive channel is distinct
8 from the send channel;
9 means for establishing a first TCP connection from a client on the client side to the first

10 component;
11 means for establishing a second TCP connection from the second component to a target
12 server on the remote side;
13 means for transmitting client-initiated TCP requests from the client to the target server by
14 packaging the client-initiated requests into HTTP messages which are transmitted on the send
15 channel; and
16 means for transmitting server-initiated TCP requests from the target server to the client by
17 packaging the server-initiated requests into HTTP messages which are transmitted on the receive
18 channel.

1 Claim 11 (previously presented): The system according to Claim 10, wherein the means for
2 transmitting client-initiated TCP requests further comprises:
3 means for receiving a client-initiated TCP request from the client at the first component
4 on the first TCP connection;
5 means for packaging the received client-initiated TCP request in an HTTP POST request
6 message;
7 means for sending the HTTP POST request message to the second component on the send
8 channel;
9 means for receiving the sent HTTP POST request message at the second component;
10 means for extracting the client-initiated TCP request from the received HTTP POST
11 request message; and
12 means for forwarding the extracted client-initiated TCP request to the target server on the

13 second TCP connection.

1 Claim 12 (previously presented): The system according to Claim 11, wherein the means for
2 transmitting client-initiated TCP requests further comprises means for acknowledging the HTTP
3 POST request by sending an HTTP POST response from the second component to the first
4 component on the send channel.

1 Claim 13 (original): The system according to Claim 12, wherein the means for establishing the
2 send channel operates in response to the means for receiving the client-initiated TCP request, and
3 wherein the means for transmitting client-initiated TCP requests further comprises:
4 means for receiving the HTTP POST response at the first component; and
5 means for closing the send channel, responsive to operation of the means for receiving the
6 HTTP POST response.

1 Claim 14 (previously presented): The system according to Claim 10, wherein the means for
2 transmitting server-initiated TCP requests further comprises:
3 means for sending an HTTP GET request message from the first component to the second
4 component on the receive channel;
5 means for receiving the sent HTTP GET request message at the second component;
6 means for receiving a server-initiated TCP request from the target server at the second
7 component on the second TCP connection;
8 means for packaging the received server-initiated TCP request in an HTTP GET response

Serial No. 09/619,178

-8-

Docket RSW9-2000-0054-US1

9 message which acknowledges the received HTTP GET request message;

10 means for sending the HTTP GET response message from the second component to the
11 first component on the receive channel;

12 means for receiving the sent HTTP GET response message at the first component;

13 means for extracting the server-initiated TCP request from the received HTTP GET
14 response message; and

15 means for forwarding the extracted server-initiated TCP request to the client on the first
16 TCP connection.

1 Claim 15 (original): The system according to Claim 14, wherein the means for transmitting
2 server-initiated TCP requests further comprises:

3 means for performing a read operation on the second TCP connection, responsive to
4 operation of the means for receiving the sent HTTP GET request message and prior to operation
5 of the means for receiving the server-initiated TCP request; and

6 means for using the received server-initiated TCP request as a result of the read operation,
7 thereby triggering operation of the means for packaging the received server-initiated TCP request
8 in the HTTP GET response message.

1 Claim 16 (original): The system according to Claim 14, wherein the means for transmitting
2 server-initiated TCP requests further comprises means for preparing to receive another server-
3 initiated TCP request by triggering operation of the means for sending the HTTP GET request
4 message from the first component to the second component, responsive to operation of the means

5 for receiving the sent HTTP GET response message at the first component.

1 Claim 17 (original): The system according to Claim 11, wherein a Multi-Purpose Internet Mail
2 Extensions (MIME) type of the HTTP POST request message is set to "binary/tcp".

1 Claim 18 (original): The system according to Claim 14, wherein a Multi-Purpose Internet Mail
2 Extensions (MIME) type of the HTTP GET request message is set to "binary/tcp".

1 Claim 19 (previously presented): A method for sending Transmission Control Protocol (TCP)
2 messages through HyperText Transfer Protocol (HTTP) systems, comprising the steps of:
3 establishing a send channel from a first component on a client side of a network, through
4 one or more HTTP-based systems, to a second component on a remote side of the network;
5 establishing a receive channel from the first component, through the one or more HTTP-
6 based systems, to the second component, wherein the receive channel is distinct from the send
7 channel;
8 establishing a first TCP connection from a client on the client side to the first component;
9 establishing a second TCP connection from the second component to a target server on
10 the remote side;
11 transmitting client-initiated TCP requests from the client to the target server by packaging
12 the client-initiated requests into HTTP messages which are transmitted on the send channel; and
13 transmitting server-initiated TCP requests from the target server to the client by
14 packaging the server-initiated requests into HTTP messages which are transmitted on the receive

15 channel.

1 Claim 20 (previously presented): The method according to Claim 19, wherein the step of
2 transmitting client-initiated TCP requests further comprises the steps of:
3 receiving a client-initiated TCP request from the client at the first component on the first
4 TCP connection;
5 packaging the received client-initiated TCP request in an HTTP POST request message;
6 sending the HTTP POST request message to the second component on the send channel;
7 receiving the sent HTTP POST request message at the second component;
8 extracting the client-initiated TCP request from the received HTTP POST request
9 message; and
10 forwarding the extracted client-initiated TCP request to the target server on the second
11 TCP connection.

1 Claim 21 (previously presented): The method according to Claim 20, wherein the step of
2 transmitting client-initiated TCP requests further comprises the step of acknowledging the HTTP
3 POST request by sending an HTTP POST response from the second component to the first
4 component on the send channel.

1 Claim 22 (original): The method according to Claim 21, wherein the step of establishing the
2 send channel operates in response to the step of receiving the client-initiated TCP request, and
3 wherein the step of transmitting client-initiated TCP requests further comprises the steps of:

Serial No. 09/619,178

-11-

Docket RSW9-2000-0054-US1

receiving the HTTP POST response at the first component; and
closing the send channel, responsive to receiving the HTTP POST response.

Claim 23 (previously presented): The method according to Claim 19, wherein the step of
transmitting server-initiated TCP requests further comprises the steps of:

sending an HTTP GET request message from the first component to the second
component on the receive channel;
receiving the sent HTTP GET request message at the second component;
receiving a server-initiated TCP request from the target server at the second component
on the second TCP connection;
packaging the received server-initiated TCP request in an HTTP GET response message
which acknowledges the received HTTP GET request message;
sending the HTTP GET response message from the second component to the first
component on the receive channel;
receiving the sent HTTP GET response message at the first component;
extracting the server-initiated TCP request from the received HTTP GET response
message; and
forwarding the extracted server-initiated TCP request to the client on the first TCP
connection.

Claim 24 (original): The method according to Claim 23, wherein the step of transmitting server-
initiated TCP requests further comprises the steps of:

3 performing a read operation on the second TCP connection, responsive to receiving the
4 sent HTTP GET request message and prior to receiving the server-initiated TCP request; and
5 using the received server-initiated TCP request as a result of the read operation, thereby
6 triggering the step of packaging the received server-initiated TCP request in the HTTP GET
7 response message.

1 Claim 25 (original): The method according to Claim 23, wherein the step of transmitting server-
2 initiated TCP requests further comprises the step of preparing to receive another server-initiated
3 TCP request by triggering the step of sending the HTTP GET request message from the first
4 component to the second component, responsive to receiving the sent HTTP GET response
5 message at the first component.

1 Claim 26 (original): The method according to Claim 20, wherein a Multi-Purpose Internet Mail
2 Extensions (MIME) type of the HTTP POST request message is set to "binary/tcp".

1 Claim 27 (original): The method according to Claim 23, wherein a Multi-Purpose Internet Mail
2 Extensions (MIME) type of the HTTP GET request message is set to "binary/tcp".

1 Claim 28 (previously presented): A method for transporting bi-directional protocol traffic
2 through uni-directional protocol systems, comprising the steps of:
3 establishing a send channel from a first component on a client side of a network
4 connection, through one or more uni-directional protocol-based systems, to a second component

5 on a remote side of the network connection;

6 establishing a receive channel from the first component, through the one or more uni-
7 directional protocol-based systems, to the second component, wherein the receive channel is
8 distinct from the send channel;

9 establishing a first bi-directional protocol connection from a client on the client side to
10 the first component;

11 establishing a second bi-directional protocol connection from the second component to a
12 target server on the remote side;

13 transmitting client-initiated bi-directional protocol requests from the client to the target
14 server by packaging the client-initiated bi-directional protocol requests into uni-directional
15 protocol messages which are transmitted on the send channel; and

16 transmitting server-initiated bi-directional protocol requests from the target server to the
17 client by packaging the server-initiated bi-directional protocol requests into uni-directional
18 protocol messages which are transmitted on the receive channel.

1 Claim 29 (previously presented): The method according to Claim 28, wherein the step of
2 transmitting client-initiated bi-directional protocol requests further comprises the steps of:

3 receiving a client-initiated bi-directional protocol request from the client at the first
4 component on the first bi-directional protocol connection;

5 packaging the received client-initiated bi-directional protocol request in a uni-directional
6 protocol write request message;

7 sending the uni-directional protocol write request message to the second component on

8 the send channel;
9 receiving the sent uni-directional protocol write request message at the second
10 component;
11 extracting the client-initiated bi-directional protocol request from the received uni-
12 directional protocol write request message; and
13 forwarding the extracted client-initiated bi-directional protocol request to the target server
14 on the second bi-directional protocol connection.

1 Claim 30 (previously presented): The method according to Claim 28, wherein the step of
2 transmitting server-initiated bi-directional protocol requests further comprises the steps of:
3 sending a uni-directional protocol read request message from the first component to the
4 second component on the receive channel;
5 receiving the sent uni-directional protocol read request message at the second component;
6 receiving a server-initiated bi-directional protocol request from the target server at the
7 second component on the second bi-directional protocol connection;
8 packaging the received server-initiated bi-directional protocol request in a uni-directional
9 protocol read response message which acknowledges the received uni-directional protocol read
10 request message;
11 sending the uni-directional protocol read response message from the second component to
12 the first component on the receive channel;
13 receiving the sent uni-directional protocol read response message at the first component;
14 extracting the server-initiated bi-directional protocol request from the received uni-

15 directional protocol read response message; and
16 forwarding the extracted server-initiated bi-directional protocol request to the client on
17 the first bi-directional protocol connection.

1 Claim 31 (new): A system for providing bi-directional messaging over uni-directional protocol
2 systems, comprising:

3 a send channel established from a first component on a client side of a network
4 connection, through at least one uni-directional protocol-based system, to a second component on
5 a remote side of the network connection;

6 a receive channel established from the first component, through the at least one uni-
7 directional protocol-based system, to the second component, wherein the receive channel is
8 distinct from the send channel;

9 a first bi-directional protocol connection established between a client on the client side
10 and the first component; and

11 a second bi-directional protocol connection established between the second component
12 and a server on the remote side;

13 wherein the first component packages client-initiated bi-directional protocol requests,
14 which are sent from the client on the first bi-directional protocol connection and received at the
15 first component, into uni-directional protocol messages and forwards the packaged client-
16 initiated protocol requests to the second component using the send channel and upon receipt of
17 the forwarded client-initiated requests, the second component extracts the client-initiated bi-
18 directional protocol requests and forwards the extracted client-initiated bi-directional protocol

19 requests to the server on the second bi-direction protocol connection, thereby providing client-to-
20 server messaging through the at least one uni-directional protocol-based system; and
21 wherein the second component packages server-initiated bi-directional protocol requests,
22 which are sent from the server on the second bi-directional protocol connection and received at
23 the second component, into uni-directional protocol messages and forwards the packaged server-
24 initiated protocol requests to the first component using the receive channel and upon receipt of
25 the forwarded server-initiated requests, the first component extracts the server-initiated bi-
26 directional protocol requests and forwards the extracted server-initiated bi-directional protocol
27 requests to the client on the first bi-direction protocol connection, thereby providing server-to-
28 client messaging through the at least one uni-directional protocol-based system.